IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Abbey, et al.	Docket No.:	ROC920030219US1
Serial No.:	10/624,352	Group Art Unit:	2188
Filed:	07/22/03	Examiner:	DOAN, DUC T.

For: APPARATUS AND METHOD FOR AUTONOMICALLY DETECTING RESOURCES IN A LOGICALLY PARTITIONED COMPUTER SYSTEM

APPEAL BRIEF

Mail Stop APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir/Madam:

This appeal is taken from the Examiner's final rejection, set forth in the Office Action dated 02/6/06, of applicants' claims 7-9, 11-14, 21-23, 25-28, 37, 40-41, 43-44, and 47-48. Applicants' Notice of Appeal under 37 C.F.R. § 1.191 was filed on 05/04/06.

REAL PARTY IN INTEREST

International Business Machines Corporation is the Real Party in Interest.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences for this patent application.

STATUS OF CLAIMS

As filed, this case included claims 1-48. In an amendment filed 11/29/05, claims 1-6, 10, 15-20, 24, 29-36, 38, 39, 42, 45 and 46 were cancelled, and claims 7, 21, 37 and 44 were amended. In the pending final office action, claims 37, 40-41, 44 and 47-48 were rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter. Claims 7-9, 11-14, 21-23, 25-28, 37, 40-41, 43-44 and 47-48 were rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent Application Publication US 2002/0052914 to Zalewski *et al.* (hereinafter "Zalewski"). The claims remaining in the case are claims 7-9, 11-14, 21-23, 25-28, 37, 40-41, 43-44, and 47-48, all of which stand finally rejected. No claim has been allowed.

STATUS OF AMENDMENTS

In response to the first office action, an amendment was filed on 11/29/05. Therefore, the claims at issue in this appeal are the claims as amended by the amendment filed 11/29/05.

SUMMARY OF CLAIMED SUBJECT MATTER

Claim 7 recites: an apparatus comprising: at least one processor (FIG. 1, 110; p. 9 lines 17-19); a memory coupled to the at least one processor (FIG. 1, 120; p. 6 lines 13-15); a plurality of logical partitions defined on the apparatus (FIGS. 1 and 4, 125A, . . ., 125N; p. 6 lines 13-17); a persistent resource database residing in the memory, the resource database including a list of resources owned by each of the plurality of logical partitions, where the resources were detected in previous power on cycles of the apparatus (FIGS. 1 and 4, 124; p. 6 lines 19-21); and a resource detection mechanism residing in the memory and executed by the at least one processor, the resource detection mechanism determining from the resource database a set of required resources owned by a selected

logical partition, detecting each resource as the resource is initialized, detecting when at least one required resource for the selected logical partition is not powered up, initiating power up of the at least one required resource that is not powered up, and starting the selected logical partition when all required resources owned by the selected logical partition have been detected (FIGS. 1 and 4, 122; p. 6 line 18 to p. 7 line 7).

Claim 12 recites: an apparatus comprising: at least one processor (FIG. 1, 110; p. 9 lines 17-19); a memory coupled to the at least one processor (FIG. 1, 120; p. 6 lines 13-15); a plurality of logical partitions defined on the apparatus (FIGS. 1 and 4, 125A, . . ., 125N; p. 6 lines 13-17); a persistent resource database residing in the memory, the resource database including a list of resources owned by each of the plurality of logical partitions, where the resources were detected in previous power on cycles of the apparatus (FIGS. 1 and 4, 124; p. 6 lines 19-21); and a resource detection mechanism residing in the memory and executed by the at least one processor, the resource detection mechanism determining from the resource database a set of required resources owned by a selected logical partition, detecting each resource as the resource is initialized, and starting the selected logical partition when all required resources owned by the selected logical partition have been detected, the resource detection mechanism further detecting when at least one required resource for a selected logical partition is not powered up and initiating power up of the at least one required resource that is not powered up (FIGS. 1 and 4, 122; p. 6 line 18 to p. 7 line 7), the resource detection mechanism initiating power off of a plurality of resources owned by the selected logical partition in response to the selected logical partition being powered off (FIG. 7, 730; p. 14 lines 10-15).

Claim 21 recites: a computer-implemented method for initializing a computer system that includes a plurality of logical partitions, the method comprising the steps of: storing in a persistent resource database a list of resources owned by each of the plurality of logical partitions during previous power on cycles of the computer system (FIG. 1, 124; p. 6 lines 19-21); determining from the resource database a set of required resources

owned by a selected logical partition (FIG. 3, 350; p. 12 lines 8-9); detecting each resource as the resource is initialized (FIG. 3, 360; p. 12 lines 10-11); detecting when at least one required resource for a selected logical partition is not powered up (FIG. 3, 380; p. 12 lines 15-17); initiating power up of the at least one required resource that is not powered up (FIG. 3, 382; p. 12 lines 18-19); and starting the selected logical partition when all required resources owned by the selected logical partition have been detected (FIG. 3, 362; p. 12 lines 10-11).

Claim 26 recites: a computer-implemented method for initializing a computer system that includes a plurality of logical partitions, the method comprising the steps of: storing in a persistent resource database a list of resources owned by each of the plurality of logical partitions during previous power on cycles of the computer system (FIG. 1, 124; p. 6 lines 19-21); determining from the resource database a set of required resources owned by a selected logical partition (FIG. 3, 350; p. 12 lines 8-9); detecting each resource as the resource is initialized (FIG. 3, 360; p. 12 lines 10-11); starting the selected logical partition when all required resources owned by the selected logical partition have been detected (FIG. 3, 362; p. 12 lines 10-11); detecting when at least one required resource for a selected logical partition is not powered up (FIG. 3, 380; p. 12 lines 15-17); initiating power up of the at least one required resource that is not powered up (FIG. 3, 382; p. 12 lines 18-19); and initiating power off of a plurality of resources owned by the selected logical partition in response to the selected logical partition being powered off (FIG. 7, 730; p. 14 lines 14-15).

Claim 37 recites: a program product comprising: (A) a resource detection mechanism that determines from a persistent resource database in a computer system that includes a plurality of logical partition a set of required resources owned by a selected logical partition, the resource detection mechanism detecting each resource as the resource is initialized and starting the selected logical partition when all required resources owned by the selected logical partition have been detected, wherein the

resource detection mechanism detects when at least one required resource for the selected logical partition is not powered up, and initiates power up of the at least one required resource that is not powered up (FIGS. 1 and 4, 122; p. 6 line 18 to p. 7 line 7); and (B) recordable computer readable signal bearing media bearing the resource detection mechanism (FIG. 1, 195; p. 6 lines 9-10).

Claim 44 recites: a program product comprising: (A) a resource detection mechanism that determines from a resource database in a computer system that includes a plurality of logical partitions a set of required resources owned by a selected logical partition, the resource detection mechanism detecting each resource as the resource is initialized and starting the selected logical partition when all required resources owned by the selected logical partition have been detected, the resource detection mechanism further detecting when at least one required resource for a selected logical partition is not powered up and initiating power up of the at least one required resource that is not powered up (FIGS. 1 and 4, 122; p. 6 line 18 to p. 7 line 7), the resource detection mechanism initiating power off of a plurality of resources owned by the selected logical partition in response to the selected logical partition being powered off (FIG. 7, 730; p. 14 lines 10-15); and (B) recordable computer readable signal bearing media bearing the resource detection mechanism (FIG. 1, 195; p. 6 lines 9-10).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The following grounds of rejection are presented for review on this Appeal:

1. **Whether claims 37, 40-41, 44, and 47-48 are unpatentable under 35 U.S.C. §101**

2. **Whether claims 7-9, 11-14, 21-23, 25-28, 37, 40-41, 43-44 and 47-88 are anticipated under 35 U.S.C. §102(b) by Zalewski**

ARGUMENT

Issue 1: Whether claims 37, 40-41, 44, and 47-48 are unpatentable under 35 U.S.C. §101

Claim 37

The examiner rejected claim 37 under 35 U.S.C. §101 as being directed to non-statutory subject matter because the "Examiner cannot find in the claim any recitation describing [sic] of the "instructions" that when executed, performing [sic] certain mechanism [sic]. Thus the claim language does not make the computer program product statutory."

Appellant respectfully asserts that the examiner has not applied the proper standard for patentability under 35 U.S.C. §101. The Commissioner has held for over 11 years that "computer programs embodied in a tangible medium, such as floppy diskettes, are patentable subject matter under 35 U.S.C. §101, and must be examined under 35 U.S.C. §102 and 103." In re Beauregard, 53 F3d 1583, 35 USPQ2d 1383 (Fed. Cir. 1995). The program product in claim 37 is a resource detection mechanism recited in (A) embodied in a tangible medium recited in (B). Limitation (B) recites: "recordable computer readable signal bearing media bearing the resource detection mechanism." This limitation is self-explanatory. The media claimed is recordable, such as floppy disks and CD-RW discs. The media claimed is also computer readable, meaning that a computer is capable of reading the media. The media claimed also bears a signal, namely the software claimed in (A). There is no requirement in 35 U.S.C. §101 to recited "instructions" that, when executed, perform certain steps, as suggested by the examiner. Because claim 37 recites computer software in (A) that resides in a tangible medium in (B), claim 37 defines statutory subject matter under 35 U.S.C. §101, and appellant respectfully requests that the examiner's rejection of claim 37 under 35 U.S.C. §101 be reversed.

Claims 40-41

Claims 40-41 depend on claim 37, and therefore recite statutory subject matter for the reasons stated above for claim 37.

Claim 43

The examiner did not reject claim 43 under 35 U.S.C. §101, but this appears to be a oversight on the part of the examiner. Claim 43 depends on claim 37, and therefore recites statutory subject matter for the reasons stated above for claim 37.

Claim 44

Claim 44 is similar in structure to claim 37, and defines computer software in (A) that resides on a tangible medium recited in (B). For the reasons given above for claim 37, claim 44 also defines statutory subject matter under 35 U.S.C. §101, and appellant respectfully requests the examiner's rejection of claim 44 under 35 U.S.C. §101 be reversed.

Claims 47 and 48

Claims 47-48 depend on claim 44, and therefore recite statutory subject matter for the reasons stated above for claim 44.

Issue 2: Whether claims 7-9, 11-14, 21-23, 25-28, 37, 40-41, 43-44 and 47-88 are anticipated under 35 U.S.C. §102(b) by Zalewski

<u>Claims 7-9, 21-23, 37 and 40-41</u>

In the final office action, the examiner rejected claim 7 as being anticipated by Zalewski. In rejecting claim 7, the examiner states that Zalewski describes the resources are monitored to determine when they are powered up, the statutes are then recorded as available for the particular partition in the HRPB structure, citing paragraph 57 of Zalewski. It is interesting to note that the examiner has maintained the rejection of claim 7 based on Zalewski in the final office action, but has shifted rationale. In the first office action, the examiner cites the turning on and off of CPUs in paragraph [0007] of Zalewski on the limitations in former claim 10, which are now in claim 7. In the final office action, the examiner cites paragraph [0057] of Zalewski instead of paragraph [0007].

In the Response to Arguments section of the final office action, the examiner states:

> Zalewski further describes the resources are monitored to determine when they are powered up, the statutes are then recorded as available for the particular partition in the HRPB [sic] structure; paragraph 57. Thus again, Zalewski taught in above paragraphs the claim's limitation of "detecting when at least one required resource for the selected logical partition is not powered up, initiating power up of the at least one required resource that is not powered up, and starting the logical partition when all required resources owned by the selected logical partition have been detected."

The examiner's poor grammar and choice of words makes it difficult to decipher the language above. There are no "statutes" referenced in Zalewski, so it is completely unclear what the examiner is referring to by "the statutes are then recorded as available for the particular partition . . .". A detailed evaluation of paragraph 57 of Zalewski will

demonstrate that the examiner's rejection of claim 7 under 35 U.S.C. §102(b) does not have merit.

Paragraph 57 of Zalewski states:

> [0057] Each HWRPB which is created by the console program will contain a CPU slot-specific database for each CPU that is in the system, or that can be added to the system without powering the entire system down. Each CPU that is physically present will be marked "present", but only CPUs that will initially execute in a specific partition will be marked "available" in the HWRPB for the partition. The operating system instance running on a partition will be capable of recognizing that a CPU may be available at some future time by a present (PP) bit in a per-CPU state flag fields of the HWRPB, and can build data structures to reflect this. When set, the available (PA) bit in the per-CPU state flag fields indicates that the associated CPU is currently associated with the partition, and can be invited to join SMP operation.

Unfortunately, there are no figures in Zalewski that show the CPU slot-specific database for each CPU that is in the system or can be added to the system without powering down the entire system. A reasonable reading of the first sentence in the paragraph quoted above is that the CPU slot-specific database includes all CPUs in the system or that can be added to the system without powering the entire system down. This means the CPU database lists all CPUs, even those that are available to other partitions. If we look at the example in FIG. 10 of Zalewski, the CPU database for Instance A might look like the following:

0		1		2		3		4		5		6		7		8		9		10		11	
PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA
1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

The PP (processor present) bit indicates whether the CPU is present in the system, while the PA (processor available) bit indicates whether the CPU is available for the logical partition that includes the CPU database. According to FIG. 10, all 12 CPUs 0-11 are available in the system. Of these, CPUs 0-3 are available to Instance A, CPUs 4-7 are available to Instance B, and CPUs 8-11 are available to Instance C. Thus, the CPU database for Instance A as shown above indicates all are present (all PP bits are set to 1), while only CPUs 0-3 are available to Instance A (PA bits for 0, 1, 2 and 3 are set to 1). The rest of the CPUs 4-11 are not available to Instance A, as shown by the PA bits of CPUs 4-11 being set to zero.

For the example in FIG. 10 of Zalewski, the CPU database for Instance B might look like the following:

0		1		2		3		4		5		6		7		8		9		10		11	
PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA
1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0

The CPU database for Instance B as shown above indicates all are present (all PP bits are set to 1), while only CPUs 4-7 are available to Instance B (PA bits for 4, 5, 6 and 7 are set to 1). The rest of the CPUs 0-3 and 8-11 are not available to Instance B, as shown by the PA bits of CPUs 0-3 and 8-11 being set to zero.

For the example in FIG. 10 of Zalewski, the CPU database for Instance C might look like the following:

0		1		2		3		4		5		6		7		8		9		10		11	
PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1

The CPU database for Instance C as shown above indicates all are present (all PP bits are set to 1), while only CPUs 8-11are available to Instance C (PA bits for 8, 9, 10 and 11 are set to 1). The rest of the CPUs 0-7 are not available to Instance C, as shown by the PA bits of CPUs 0-7 being set to zero.

Now we consider what happens when CPUs 2 and 3 are switched from Instance A to Instance B, and CPUs 8 and 9 are switched from Instance C to Instance B, as shown in the example in FIG. 13 of Zalewski. The CPU database for Instance A after the transfer might look like the following:

0		1		2		3		4		5		6		7		8		9		10		11	
PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA
1	1	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

The "available" bits (PA) for CPUs 2 and 3 are set to zero for Instance A because they are no longer available to Instance A -- they have been transferred to Instance B. The CPU database for Instance B after the transfer in FIG. 13 might look like the following:

0		1		2		3		4		5		6		7		8		9		10		11	
PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA
1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0

The "available" bits (PA) for CPUs 2, 3, 8 and 9 are set to one for Instance B because they have been transferred to Instance B. The CPU database for Instance C after the transfer in FIG. 13 might look like the following:

0		1		2		3		4		5		6		7		8		9		10		11	
PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA	PP	PA
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1

The "available" bits (PA) for CPUs 8 and 9 are set to zero for Instance C because they are no longer available to Instance C -- they have been transferred to Instance B.

These examples above show that the presence of a CPU as indicated by the PP bit being set to one indicates the CPU is present in the system, and may be available at some future time. Thus, in the initial CPU database for Instance B for the configuration shown in FIG. 10, the present bits PP for CPUs 0-3 and 8-11 are set to one to indicate these CPUs are in the system, and might be available to Instance B at some future time. The presence of CPUs in the system in Zalewski assumes these CPUs are powered up. Zalewski clearly teaches reallocating CPUs that are powered up between logical partitions, as shown in the example in FIG. 13 as discussed above. However, nowhere does Zalewski teach or suggest detecting when at least one required resource for the

selected logical partition is not powered up, as expressly recited in claim 7. Furthermore, Zalewski does not teach or suggest initiating power up of the at least one required resource that is not powered up, as expressly recited in claim 7. Paragraph [0057] of Zalewski, which was cited by the examiner as allegedly teaching these limitations, includes neither. As a result, claim 7 is allowable over Zalewski.

Claims 21 and 37 contain limitations similar to those in claim 1, which are addressed in detail above. As a result, claims 21 and 37 are allowable for the same reasons given above for claim 1.

Claims 8-9, 22-23 and 40-41 depend on claims 1, 31 and 37, respectively, which are allowable for the reasons given above. As a result, claims 8-9, 22-23 and 40-41 are allowable as depending on allowable independent claims.

For the many reasons given above, appellant respectfully requests the examiner's rejection of claims 7-9, 21-23, 37 and 40-41 under 35 U.S.C. §102(b) be reversed.

Claims 11-14, 25-28, 43-44 and 47-48

The arguments made above with respect to claims 7-9, 21-23, 37 and 40-41 apply equally to claims 11-14, 25-28, 43-44 and 47-48, and are incorporated herein by reference. Claim 11 recites:

> 11. The apparatus of claim 7 wherein the resource detection mechanism initiates power off of a plurality of resources owned by the selected logical partition in response to the selected logical partition being powered off.

In rejecting claim 11, the examiner states the claim is rejected based on the same rationale as in the rejection of claim 7, then goes on to state:

> Zalewski describes the CPUs logically assigned to each partition can be turned "on" and "off" dynamically (Zalewski's column 2, paragraph 11). Zalewski further describes the console program provide [sic] a mechanism to remove a resource such as a CPU from available CPUs within a partition in response to a shutdown for the instant operating system running in that partition; Zalewski's paragraph 56.

As described in the detailed example above in the discussion of claim 1, removing a CPU from available CPUs within a partition does not result in the CPU being powered off. Paragraph 11 of Zalewski indicates CPUs can be turned "on" and "off" dynamically via normal operating system operator commands without reboot. Appellant challenges the examiner to provide a single operating system operator command that allows powering down a CPU. CPUs may be enabled and disabled via operating system commands, effectively turning them "on" and "off" as discussed in Zalewski. However, these operating system commands do not power on and power off a CPU. Because the turning of CPUs on and off as discussed in Zalewski occurs with power always on, the turning of CPUs on and off in Zalewski does not read on initiating power off of a plurality of resources owned by the selected logical partition, as recited in claim 11. For these reasons, claim 11 is allowable over Zalewski.

Claims 25 and 43 include limitations similar to those in claim 11 discussed above. As a result, claims 25 and 43 are allowable for the same reasons given above for the allowability of claim 11.

Claims 12, 26 and 44 include limitations similar to those discussed in claim 11above. As a result, these claims are allowable for the same reasons given above for the allowability of claim 11.

Claims 13-14, 27-28 and 47-48 depend on claims 12, 26 and 44, which are allowable for the reasons given above. As a result, claims 13-14, 27-28 and 47-48 are allowable as depending on allowable independent claims.

CONCLUSION

Claims 7-9, 11-14, 21-23, 25-28, 37, 40-41, 43-44, and 47-48 are addressed in this Appeal. For the numerous reasons articulated above, applicants maintain that the rejection of claims 7-9, 11-14, 21-23, 25-28, 37, 40-41, 43-44, and 47-48 under 35 U.S.C. §§101 and 102(b) is erroneous.

Applicants respectfully submit that this Appeal Brief fully responds to, and successfully contravenes, every ground of rejection and respectfully requests that the final rejection be reversed and that all claims in the subject patent application be found allowable.

Respectfully submitted,

/derekpmartin/
Derek P. Martin
Reg. No. 36,595

MARTIN & ASSOCIATES, L.L.C.
P.O. Box 548
Carthage, MO 64836-0548
(417) 358-4700
Fax (417) 358-5757

CLAIMS APPENDIX

1-6 (Cancelled)

7. An apparatus comprising:

at least one processor;

a memory coupled to the at least one processor;

a plurality of logical partitions defined on the apparatus;

a persistent resource database residing in the memory, the resource database including a list of resources owned by each of the plurality of logical partitions, where the resources were detected in previous power on cycles of the apparatus; and

a resource detection mechanism residing in the memory and executed by the at least one processor, the resource detection mechanism determining from the resource database a set of required resources owned by a selected logical partition, detecting each resource as the resource is initialized, detecting when at least one required resource for the selected logical partition is not powered up, initiating power up of the at least one required resource that is not powered up, and starting the selected logical partition when all required resources owned by the selected logical partition have been detected.

8. The apparatus of claim 7 wherein the resources include at least one hardware resource.

9. The apparatus of claim 7 wherein the resources include at least one software resource.

10 (Cancelled)

11. The apparatus of claim 7 wherein the resource detection mechanism initiates power off of a plurality of resources owned by the selected logical partition in response to the selected logical partition being powered off.

12. An apparatus comprising:

at least one processor;

a memory coupled to the at least one processor;

a plurality of logical partitions defined on the apparatus;

a persistent resource database residing in the memory, the resource database including a list of resources owned by each of the plurality of logical partitions, where the resources were detected in previous power on cycles of the apparatus; and

a resource detection mechanism residing in the memory and executed by the at least one processor, the resource detection mechanism determining from the resource database a set of required resources owned by a selected logical partition, detecting each resource as the resource is initialized, and starting the selected logical partition when all required resources owned by the selected logical partition have been detected, the resource detection mechanism further detecting when at least one required resource for a selected logical partition is not powered up and initiating power up of the at least one required resource that is not powered up, the resource detection mechanism initiating power off of a plurality of resources owned by the selected logical partition in response to the selected logical partition being powered off.

13. The apparatus of claim 12 wherein the resources include at least one hardware resource.

14. The apparatus of claim 12 wherein the resources include at least one software resource.

15-20 (Cancelled)

21. A computer-implemented method for initializing a computer system that includes a plurality of logical partitions, the method comprising the steps of:

storing in a persistent resource database a list of resources owned by each of the plurality of logical partitions during previous power on cycles of the computer system;

determining from the resource database a set of required resources owned by a selected logical partition;

detecting each resource as the resource is initialized;

detecting when at least one required resource for a selected logical partition is not powered up;

initiating power up of the at least one required resource that is not powered up; and

starting the selected logical partition when all required resources owned by the selected logical partition have been detected.

22. The method of claim 21 wherein the resources include a hardware resource.

23. The method of claim 21 wherein the resources include a software resource.

24. (Cancelled)

25. The method of claim 21 further comprising the step of initiating power off of a plurality of resources owned by the selected logical partition in response to the selected logical partition being powered off.

26. A computer-implemented method for initializing a computer system that includes a plurality of logical partitions, the method comprising the steps of:

storing in a persistent resource database a list of resources owned by each of the plurality of logical partitions during previous power on cycles of the computer system;

determining from the resource database a set of required resources owned by a selected logical partition;

detecting each resource as the resource is initialized;

starting the selected logical partition when all required resources owned by the selected logical partition have been detected;

detecting when at least one required resource for a selected logical partition is not powered up;

initiating power up of the at least one required resource that is not powered up; and

initiating power off of a plurality of resources owned by the selected logical partition in response to the selected logical partition being powered off.

27. The method of claim 26 wherein the resources include a hardware resource.

28. The method of claim 26 wherein the resources include a software resource.

29-36 (Cancelled)

37. A program product comprising:

(A) a resource detection mechanism that determines from a persistent resource database in a computer system that includes a plurality of logical partition a set of required resources owned by a selected logical partition, the resource detection mechanism detecting each resource as the resource is initialized and starting the selected logical partition when all required resources owned by the selected logical partition have been detected, wherein the resource detection mechanism detects when at least one required resource for the selected logical partition is not powered up, and initiates power up of the at least one required resource that is not powered up; and

(B) recordable computer readable signal bearing media bearing the resource detection mechanism.

38. (Cancelled)

39. (Cancelled)

40. The program product of claim 37 wherein the resources include at least one hardware resource.

41. The program product of claim 37 wherein the resources include at least one software resource.

42. (Cancelled)

43. The program product of claim 37 wherein the resource detection mechanism initiates power off of a plurality of resources owned by the selected logical partition in response to the selected logical partition being powered off.

44. A program product comprising:

(A) a resource detection mechanism that determines from a resource database in a computer system that includes a plurality of logical partitions a set of required resources owned by a selected logical partition, the resource detection mechanism detecting each resource as the resource is initialized and starting the selected logical partition when all required resources owned by the selected logical partition have been detected, the resource detection mechanism further detecting when at least one required resource for a selected logical partition is not powered up and initiating power up of the at least one required resource that is not powered up, the resource detection mechanism initiating power off of a plurality of resources owned by the selected logical partition in response to the selected logical partition being powered off; and

(B) recordable computer readable signal bearing media bearing the resource detection mechanism.

45. (Cancelled)

46. (Cancelled)

47. The program product of claim 44 wherein the resources include at least one hardware resource.

48. The program product of claim 44 wherein the resources include at least one software resource.

EVIDENCE APPENDIX

An Evidence Appendix is not required for this Appeal Brief.

RELATED PROCEEDINGS APPENDIX

A Related Proceedings Appendix is not required for this Appeal Brief.